Table of Contents

[Introduction and Project Plan 2](#_Toc166672067)

[Data Understanding and Exploratory Data Analysis (EDA) 350-400 [459] 3](#_Toc166672068)

[Data Preparation 300-350 [394] 9](#_Toc166672069)

[Machine Learning Implementation and Evaluation 400-450 [416] 11](#_Toc166672070)

[5. Results, Discussion, and Recommendations 300 13](#_Toc166672071)

[Conclusion 17](#_Toc166672072)

[References 17](#_Toc166672073)

# 

# Introduction and Project Plan

There were two main objectives to this project

1. Use machine learning and exploratory data analysis (EDA) to predict late deliveries,
2. Identify distinct customer segments.

These objectives are important to the practical needs of supply chain operations where on time deliveries are crucial for maintaining customer satisfaction and optimizing logistical operations. Predicting late deliveries can significantly enhance operational efficiency and reduce costs associated with delayed shipments (He & Ma, 2013). Similarly, understanding customer segments can guide marketing strategies and improved customer service, which leads to customer retention and acquisition (Liu & Ji, 2019).

**Project Management and Planning**

To manage this project properly, I used a structured approach using the Agile methodology as described in Advances in Computers(Zelkowitz, 2002), focusing on flexibility and slow, building progress through regular sprints. The timeline spanned 2 weeks, broken down as follows in the below fig 0.1.

****

*Figure 0.1 Project Timeline*

The project’s progress was monitored with daily reviews of the previous day’s work. I would check my goals, milestones and what I got done/had to do and kept track in Notion, a productivity and notebook tool, to ensure that the project milestones were being met.

I think this plan is a comprehensive approach to tackling the project’s objectives. It allows systematic progress through each phase while adapting to any challenges or changes in project scope.

# Data Understanding and Exploratory Data Analysis

The dataset chosen for this capstone project is from DataCo Global, a dataset I found on Kaggle with a lot of supply chain data, with 53 features and 180,519 rows (Link in  [References)](#_References) . This big dataset provides a great start to addressing critical challenges in the supply chain management world.

It had a lot of useful features with geographical features such co-ordinates, city, state and country features as well as many features which described customer buying behaviour.

The data varies a lot in how skewed each feature is as seen in fig 0.2.

A screenshot of a computer

Description automatically generated

*Figure 0.2 Statistical description of Dataset*

From initial findings based on my confusion matrix, profile report and pair plot I can see some correlations between late deliveries (late\_delivery\_risk) and a lot of geographical data and shipping methods and scheduling as seen in Figure 1.0 and 1.1. I intentionally am ignoring correlations which could include any data leakage which I will discuss further in the [next section](#_Data_Preparation_300-350).

A screenshot of a computer generated image

Description automatically generated *A screenshot of a computer screen

Description automatically generated*

*Figure 1.0 correlation Matrix of Dataset Figure 1.1 Correlation Table*

With late delivery correlations being mainly centred around geographical features I decided to plot 2 maps using MapBox to better visualise this. I had to create an account to get an access token before creating 2 live and interactive “open-street-map” style maps which showed either late deliveries only (*Figure 1.3)* or late deliveries vs on time deliveries(*Figure 1.4)* across the globe.

*A map of the world

Description automatically generated ­­­A map of the world with yellow dots

Description automatically generated*



*Figure 1.3 Late Deliveries Figure 1.4 Late vs on time deliveries*

To get a clearer understanding of the geographical features that contribute to late delivery I plotted out the following graphs;

Figs 1.5, 1.6 & 1.7 shows how our deliveries performed with other geographical features such as markets top ten states and top ten cities.

A graph of different colored bars

Description automatically generated

*Figure (1.5) Late vs On-time Deliveries by Market*

A graph of different colored bars

Description automatically generated

*Figure (1.6) Late vs On-time Deliveries by State*

A graph of different colored bars

Description automatically generated

*Figure (1.7) Late vs On-time Deliveries by State*

The following figure 1.8 is a graph showing Late vs On-time Deliveries by our 4 Shipping Modes; Standard, First class, Second Class and Same day.

A graph of different colored bars

Description automatically generated with medium confidence

*Figure 1.8 Late vs On-time Delivery by Shipping Mode*

As mentioned already, geographical features are not the only correlators with late delivery so next I split our “sales \_per\_customer” feature into bins of 50’s and compared them to late delivery vs on-time delivery in Fig 2.0 and beside is a line chart tracking the counts in fig 2.1

A graph of sales

Description automatically generatedA graph with lines and dots

Description automatically generated

*Figure 2.0 Delivery status by sales range Figure 2.1 Line Chart of fig 2.0*

Lastly I plotted some features to do with scheduling against late vs on-time deliveries. First being Late vs On-time Deliveries by Days Scheduled for Shipping time in fig 3.0. Which showed an interesting negative linear relationship between tardiness and more days scheduled for shipping

A graph of a number of different colored bars

Description automatically generated with medium confidence

*Figure 3.0 Late vs On-time Deliveries by Days Scheduled for Shipping time*

In the following graphs I had to first find the min max for shipping dates, create year indexes for the four years present in the dataset then map them onto the df before creating 4 quarters for each of those years, and then creating a quarter index which labeled each of the quarters of the years individually. Then I overlayed a line chart over each of these barcharts which I think helped to visluase the results better, in the following figure 3.1 / 3.2 / 3.3 / 3.4

*A graph of a bar chart

Description automatically generated with medium confidence*

*Figure 3.1 Late vs On-time Deliveries by Year* *A graph with a line going up

Description automatically generated with medium confidence*

*Figure 3.2 Line Chart of 3.1*

*A graph of sales and delivery

Description automatically generated* *A graph with lines and dots

Description automatically generated*

*Figure 3.3 Late vs On-time Deliveries by Quarter Figure 3.4 Line Chart of 3.3*

# Data Preparation

My data pre-processing process began with importing a series of libraries from sklearn, matplotlib, pandas, seaborn, plotly, yellowbrick as well as setting my column width to max to get a full picture of my dataset (Figure 4.0).

A screenshot of a computer program

Description automatically generated

*Figure 4.0 libraries, warnings and max column width*

I then went through the laborious task of renaming all my features with the snake convention which would be worth it in the long run, seeing as how I was to be spending a fortnight with the data. I made sure to do this before messing around with the data any further so I didn’t have to map each new name onto the old name but rather just map onto the original order the features were in.

A screen shot of a computer screen

Description automatically generated

*Figure 4.1*

Following this I dopped some columns that I was sure I would have no need of in the EDA and model development phase of this project they were “customer\_email, customer\_fname', customer\_lname, customer\_password, product\_image, product\_status, order\_zipcode”.

The next step was to check for any null data and non-numerical data types with .info().

A screenshot of a computer

Description automatically generated

*Figure 4.2 df.info()*

With no null values, I went straight to work on converting the object data types into integers. I would this one of two ways, for features with small(<20) unique value counts I would just manual replace the string with a integer such as Customer Country in Figure 4.3 and for features with larger(>20) unique value counts I used enumeration to automatically plot the numbers on like customer city in Figure 4.3, making sure to keep track of the order in excel and double checking by comparing the order pre-enumeration with the .value\_counts() and post-enumeration with .value\_counts(). All of the data connecting the numbers to the strings is kept in the data dictionary supplied alongside this word document.

A screenshot of a computer program

Description automatically generated

*Figure 4.3 Converting to Numerical values*

Finally, I mentioned I wanted to prevent data leakage in the previous section when discussing geographical features. Data leakage is problem in machine learning where information from ‘outside’ the training dataset is used to create the model. This happens when data that would not be available at the time of prediction is used during the training phase, leading to overfitting (Narayanan, 2023). As you can see in my third commit on github, my models were returning 100% training and accuracy, this was due to me leaving in features that contributed to data leakage i.e features that could only be known after the fact and were useless as predictors such as delivery status.

# 

# Machine Learning Implementation

## Objective 1

A big inspiration and source of guidance for this project was “Introduction to machine learning” by Müller & Guido. I used a lot of insights in it for my model selection and development, pecifically the reference at the end of chapter two. Using it I was able to choose Gradient Boosted Classifier, kernel SVM and Gaussian Naïve Bayes (Müller, 2016). Findings and conclusions of the models will be discussed in the [next section](#_5._Results,_Discussion,).

**Gradient boosted classifier**

My Gradient boosted classifier was used with a 20% testing size, and had a random state of 0. I chose this over a random forest model for it’s speed and efficiency. I repeated the model a further 2 times this time tuning it’s max depth down to 2 and learning rate to 0.1 and then up to 4 and its learning rate 1.

**Kernel SVM**

For my SVM I standardised the data and created a pipeline, using a 20% test size I chose to work on a sample of 30% of the data because, as you can see from my GitHub commit 5 and below, the SVM was taking just far too long to load. I used a linear kernel and experimented with both the C and gamma parameters.

**Naïve Bayes**

For my Naïve Bayes I did a test of one with a standard scaler and one with a MinMax Scaler. I used a test size of 0.2 and had a random state of 0.

**Models Eval 100**

Out of these three models it’s my Gradient boosted classifier that performed the best at classifying late deliveries based on the selected features. With the higher max\_depth=4, learning\_rate=1 performing with the highest accuracy but being slightly overfitted. And the default being the best of the three Gradient boosted classifier choices.

## Objective 2

**PCA**

First I conducted PCA on my dataset, choosing to use three components I then printed out the columns with some descriptive statistics. I was able to plot out this reduced dimension in a 3D projection in Figure 5.0

A graph with a green splatter on it

Description automatically generated

*Figure 5.0 3D projection of the data in the reduced dimension*

I then used the Elbow Method before implementing my k means clustering model to find the optimal amount of clusters to use, which in this case the elbow was at 4.

A graph with lines and numbers

Description automatically generated

*Figure 5.1 Elbow method before KMeans model.*

**K means Clustering**

I used the recommended 4 clusters on a KMeans model and plotted out the results. As you can see in previous commits I initially chose to use the ‘visidis’ palette on all my models but it was not a great palette for distinguishing between clusters, I later changed it to ‘Set1’. I then plotted out both the clusters and their distribution in figs 5.2 & 5.3.

A graph with a plot of the clusters

Description automatically generated with medium confidenceA graph of a number of bars

Description automatically generated with medium confidence

*Figure 5.2 Plot of clusters Figure 5.3 Dist of Clusters*

# 5. Results

## Objective 1

**Model 1**

As we can see in table 1.0, test 3 had the highest accuracy but suffered from some slight over fitting, Overall test 1 was the best performer.

**

*Table 1.0 GradientBoostingClassifier*

**Model 2**

This model wasn’t as accurate but adjusting the C parameter adjusted the Accuracy.



*Table 1.1 Gaussian Naïve Bayes*

**Model 3**

As we can see in table 1.3, Both tests performed the same with the two scalers.



*Table 1.2 Gaussian Naïve Bayes*

In future works with this dataset I would put more emphasis on feature engineering over parameter tuning. I would also try and clearly state at the beginning of the project any clear goals or questions that want to be answered by the supply chain or marketing teams, by the EDA or models.

From the EDA we can see that late deliveries are a major problem and occur a majority of the time. There are clear correlations with shipping mode and geographical features as shown in [Section 2](#_Data_Understanding_and). An action that can be taken right away, is to just increase the days scheduled for shipping, a possibly temporary solution just until the supply chain team can source and fix whatever is causing the late deliveries.

## Objective 2

From our findings we have 4 customer segments, here are some insights about their behaviour and traits;



*Figure 6.0 Customer segments and their characteristics*

Here are the plots showing clusters as hues with two selected features;

A graph of different colored dots

Description automatically generated

*Figure 6.1 Sales vs Benefit per Order*

A graph with different colored dots

Description automatically generated

*Figure 6.2 Quantity vs Price*

A graph of a chart

Description automatically generated with medium confidence

*Figure 6.3 Month vs Sales*

A graph of different colored dots

Description automatically generated

*Figure 6.4 Latitude vs Sales*

A graph of different colored dots

Description automatically generated

*Figure 6.5 Longitude vs Sales*

A graph of different colored objects

Description automatically generated

*Figure 6.6 Order profit vs discount rate*

A graph with green and blue dots

Description automatically generated

*Figure 6.7 Sales vs Clusters*

# Conclusion

To summarise my project, I chose a dataset which had, what I believed, was of appropriate size and held all the necessary features to complete my objectives. I cleaned the data and found some interesting correlations between late deliveries and geographical, scheduling and sales features. I used three supervised classification models, did some param tuning and found boosted classifier to perform the best for Objective 1.

I then conducted PCA and used the elbow method to use an unsupervised KMeans model to find 4 customer segments and visualised some of their distinct behaviours and characteristics.

I learned a lot from doing this project, and in future works I would definitely like to explore feature engineering more to see how it influences my models.

# References

Github: Wordcount: 1916 words

Dataset: https://data.mendeley.com/datasets/8gx2fvg2k6/5

He, Z., & Ma, Y. (2013). *The necessity of using machine learning techniques in supply chain management: A literature review and insights*. ITM Web of Conferences.

Liu, Y., & Ji, R. (2019). *Customer segmentation in big data using clustering algorithms*. Journal of Big Data.

Narayanan, R (2023) *Data Leakage in Machine Learning: Detect and Minimize Risk*. Available at: https://builtin.com/machine-learning/data-leakage (Accessed: 6May 2024).

Müller, A.C. (2016) *Introduction to machine learning with python: A guide for data scientists*. Sebastopol, CA: O’Reilly Media, Inc.

Zelkowitz, M.V., 2002. 'Chapter 2'. In: *Advances in Computers*, vol. 62, London: Academic Press, p. 12.